AS049-1-REIN-WCU-V1 Wireless Communication Unit

Quick Start guide for AS049-1-REIN-WCU-V1 Wireless Communication Unit

The quick start guide for AS049-1-REIN-WCU-V1 serves a guide for user to get start with AS049-1-REIN-WCU-V1 Wireless Communication Unit. It explains step by step procedure to get started to use this unit and explains how to interface the board with external systems.

Contents

Table of Figures

1. Introduction

The Wireless Communication Unit is a communication card designed to add IoT (Internet of Things) connection to any system. For Automotive domain it serves an advantage to upgrade an existing ECU(Electronic Control Unit) for cloud connectivity. It has wireless modules such as BLE, Wi-Fi, LTE+GPS on board. Figure 1-1 shows the block diagram of the solution.

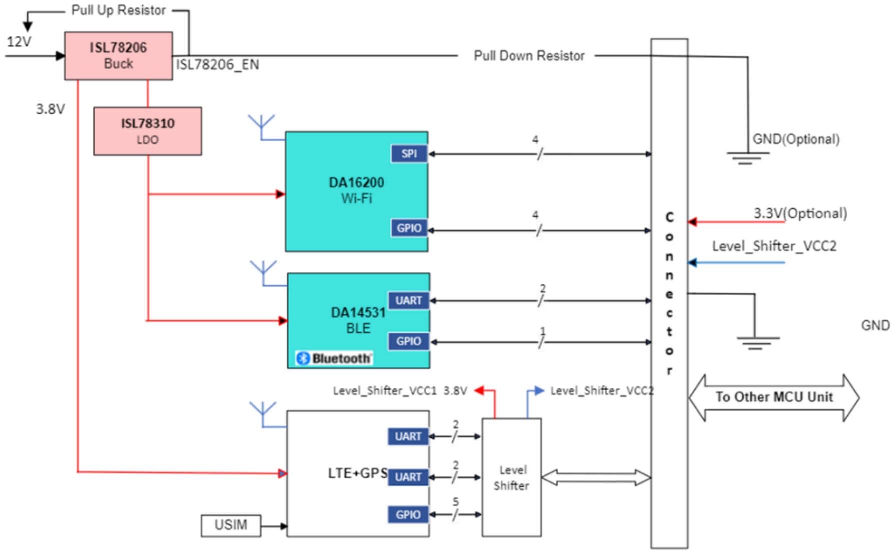

Figure 1-1 Wireless Communication Unit

2. Scope of the document

The scope of the document is to explain the AS049-1-REIN-WCU-V1 Wireless Communication Unit (WCU) hardware and guide the user to get started on the same with step-by-step procedure on operating each module on the board and connection with other systems.

3. AS049-1-REIN-WCU-V1 Hardware

The AS049-1-REIN-WCU-V1 hosts Renesas's DA16200MOD Wifi Module, DA14531MOD BLE modules and LTE+PS module from Quectel (EC25). Each module can interface to any Host MCU boards over UART, SPI and IO interfaces on J1 connector as shown in Figure 3-2.

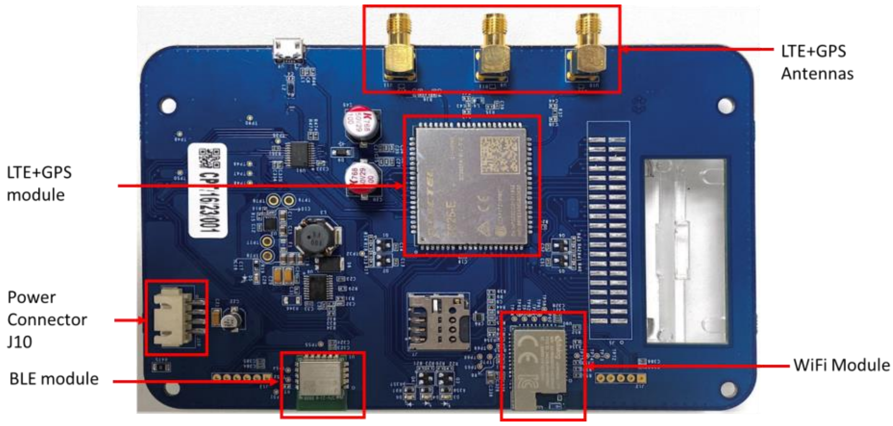

Figure 3-1. Front View of AS049-1-REIN-WCU-V1

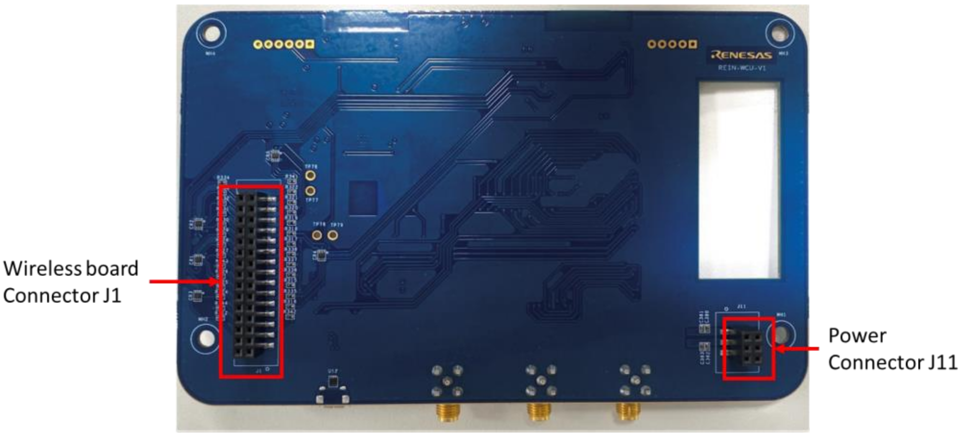

Figure 3-2 Back View

1. **Wireless Board Connector:**
   Each module on the board can interface with host MCUs via this connector. The Connector has I/Os, SPI (for WiFi module), UARTs (for BLE, LTE, GPS modules) interfaces for communication with host controllers.

2. **Power Connector J10**:
   This connector can be used to power the board individually from 12v DC supply.
   Make sure that this connector is not used when the board is powered from Power Connector J11.

3. **Power Connector J11:**
   This connector is mainly provided to the power the board using extern host controller system. In this case, J10 should not be used. For eg. AS260 can power AS049-1-REIN-WCU-V1 via this connector.

4. Getting Started with AS049-1-REIN-WCU-V1

This section provides the guidance on getting started with AS049-1-REIN-WCU-V1. Using the connectors shown in section AS049-1-REIN-WCU-V1 Hardware , this board can be interfaced with host MCU system to operate the communication modules.

4.1 Operating the board individually

As mentioned in section AS049-1-REIN-WCU-V1 Hardware the board can be powered individually using Power connector J10.

The board has following modules:

(1) DA16200MOD : Ultra low power Wi-Fi module for Battery Powered IoT Devices

(2) DA14531MOD : SmartBond TINY™ Module, world's smallest and lowest power Bluetooth 5.1 SoC

(3) EC25 : LTE+GPS module

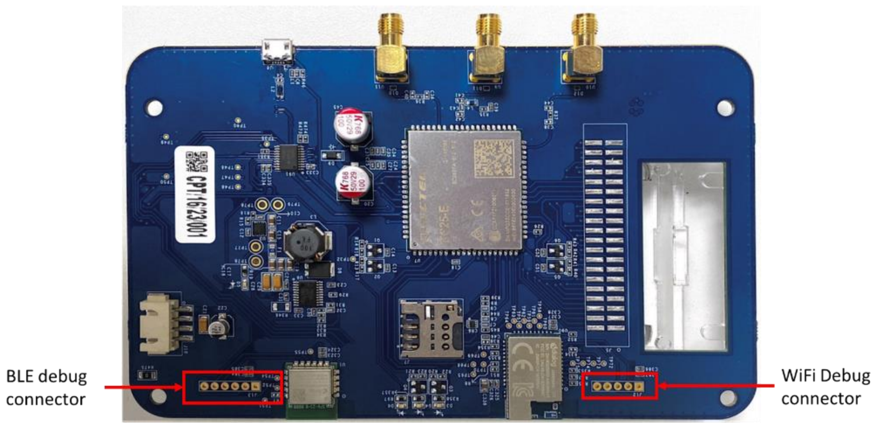

Figure 4-1 Debug connector for BLE And WiFi modules

The WiFi and BLE modules can be programmed using the debug connectors shown in Figure 4-1. EC25 module should be powered on using the power on procedure.

4.1.1. Power up the board

To power the board, apply 12v DC to the connector J10 such that the polarity is maintained as shown in below schematic screen shot:

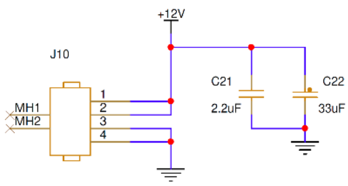

Figure 4-2 Power connector J10

Once the board is powered the power indication LED D5 should glow.

4.1.2. Power up and program Wi-Fi module.

The board hosts DA16200MOD Wi-Fi module, which is programmed with default firmware with UART communication with host controller. However, the board is designed to use Wi-Fi module in SPI mode to communicate with host. Thus it should be reprogrammed with SPI firmware which can be downloaded from here.

In this case DA16200 is SPI slave device.

After downloading and extracting the zip file, the folder will show below options:

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
DA16200_IMG_FreeRTOS_ATCMD_SPI_EVK_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,704 KB	No	1,748 KB	3%	11/7/2023 1:36 PM
DA16200_IMG_FreeRTOS_ATCMD_SPI_SPARKFUN_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,707 KB	No	1,754 KB	3%	11/10/2023 5:14 PM
DA16200_IMG_FreeRTOS_ATCMD_UART1_EVK_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,706 KB	No	1,750 KB	3%	11/7/2023 1:36 PM
DA16200_IMG_FreeRTOS_QFN_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,564 KB	No	1,608 KB	3%	11/7/2023 1:36 PM
DA16600_IMG_FreeRTOS_ATCMD_SPI_EVK_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,917 KB	No	1,965 KB	3%	11/7/2023 1:36 PM
DA16600_IMG_FreeRTOS_ATCMD_UART2_EVK_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,917 KB	No	1,966 KB	3%	11/7/2023 1:37 PM
DA16600_IMG_FreeRTOS_QFN_v3.2.8.1_4MB.zip	Compressed (zipped) Fol...	1,776 KB	No	1,824 KB	3%	11/7/2023 1:37 PM

Figure 4-3 DA16200 firmware images

Use images from 'DA16200_IMG_FreeRTOS_ATCMD_SPI_EVK_v3.2.8.0_4MB'. To download the firmware images please follow the steps below:

1. DA16200MOD is by default in sleep mode. To take it out of low power, pull up Wi-Fi_RTC_PWR_KEY_RES (pin 5 of J1) to 3.3v.

Note: While using with Host MCU, make sure to use one of MCU's GPIO to control this pin to pull it up.

2. DA16200MOD can be programmed using another UART port populated on J12 connector as below:

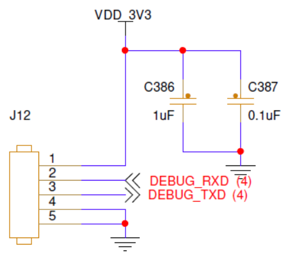

Figure 4-4 Wi-Fi Debug connector

Connect the DEBUG_RXD and DEBUG_TXD of the to connector Rx and Tx of an external USB-UART TTL module.

3. Open Tera Term and select the com port with respect to USB-UART TTL. Set the baudrate to 230400 as shown below:

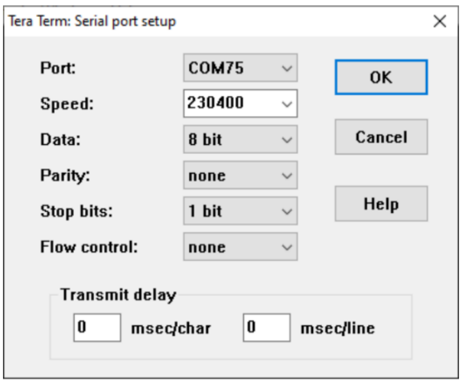

Figure 4-5 Baudrate selection for Wi-Fi debug UART

4. Once tera term is running and connected to the DA16200, open the **Control** tab, and select the **Macro** menu item.
5. When the **MACRO: Open Macro** file selection window opens, navigate to the directory where the firmware images are stored and select the .ttl file as below.

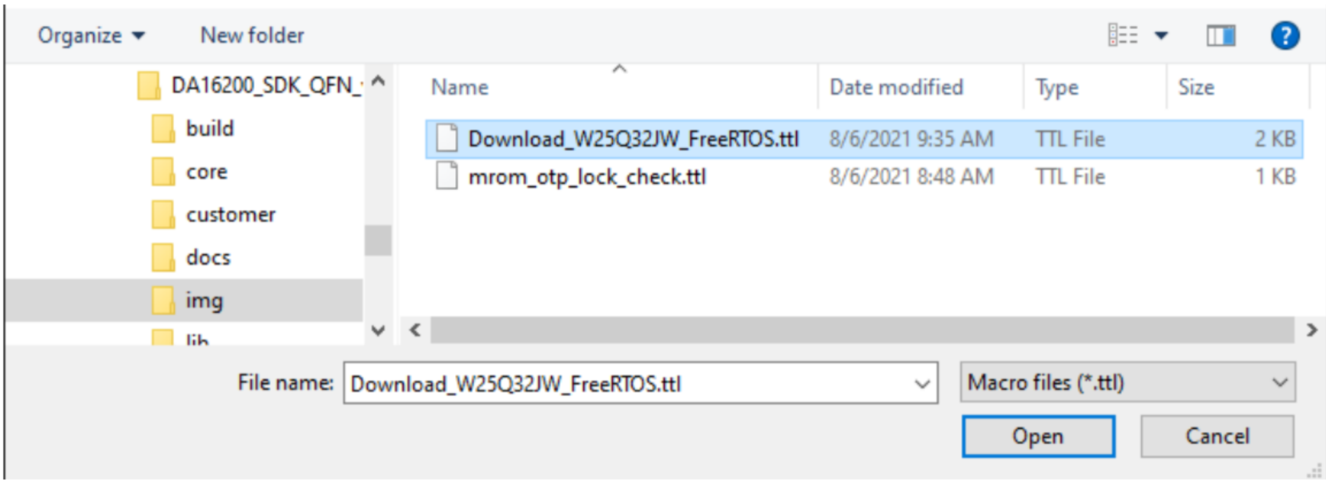

Figure 4-6 Select the .ttl file

6. After opening the macro file, the firmware image is programmed as shown in below figure.

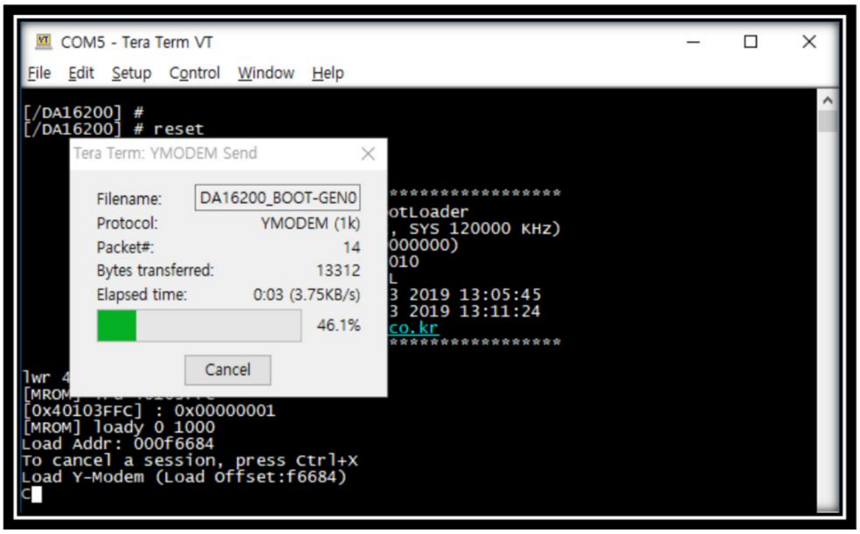

Figure 4-7 Programming the firmware.

It will program both boot and RTOS images.

7. DA16200 will reboot automatically after all images are programmed.

Now DA16200 is ready to be used with Host MCU over SPI.

Below are the Pins that should be used to communicate DA16200 with Host MCU:

Table 1 DA16200 I/O and SPI lines

Wifi I/O and SPI lines	Pins on J1
WiFi_WAKE_UP_RES	Pin 8
WiFi_INTERRUPT_RES (Interrupt to host)	Pin 10
Wi-Fi_RTC_PWR_KEY_RES	Pin 5
MSPI2SO_RES (SPI MOSI)	Pin 19
MSPI2SI_RES (SPI MISO)	Pin 21
MSPI2SC_RES (SPI clock)	Pin 23
MSPI2CSS0_RES (SPI Slave select)	Pin 22

Please refer UM-WI-003 DA16200 DA16600 Host Interface and AT Command User Manual (renesas.com) for more details on Host interface with DA16200.

4.1.3. Program BLE module

The board hosts DA14531MOD BLE module. It is bare module without firmware. It should be programmed using BLE debug connector J13 shown below:

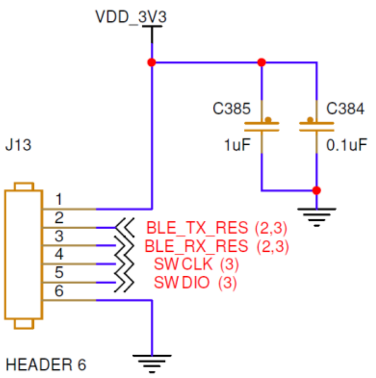

Figure 4-8 BLE debug connector

BLE module is programmed using SWD protocol. To program the module, use SWDIO and SWDCLK pins on the connector. To control BLE modules using AT commands, it should be Programmed with Codeless firmware.

User can refer 1. Introduction to CodeLess — DA145XX Tutorial SDK Getting started (renesas.com) to understand details on how to use Codeless firmware with DA14531MOD and how to send AT commands to DA14531.

The module on the board can be programmed using the SWD pins on mother board of DA14531 DK PRO or any SWD programmer and SmartBond Flash Programmer.

Once the codeless firmware is programmed BLE module is ready to use with Host controller.

The UART communication lines for BLE AT commands are populated as below:

Table 2 DA14531 Communication UART lines

BLE UART lines	Pins on J1	Pins on J13
BLE_TX_RES (2,3)	Pin16	Pin2
BLE_RX_RES (2,3)	Pin18	Pin3

4.1.4. Power up and test Quectel EC25

The board hosts Quectel EC25 LTE and GPS module. This module should be powered by a power up sequence as below:

Pull up LTE_PWR_KEY (Pin 30 of J1) for >500 ms and then pull down. Once power up sequence is done Status LED D6 should glow.

Make sure that SIM card is inserted in SIM slot and, Main and Div antennas are connected on the board as shown in below figure. Also configure the SIM as per the network provider and country region.

Once status LED is glowing user can connect EC25 to Host Controller with below communication pins:

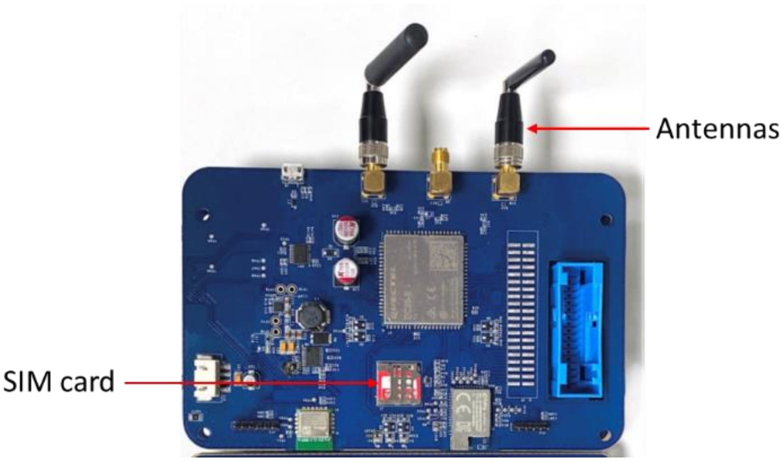

Figure 4-9 SIM card and Antennas

4.2 Operating the board with other systems

AS049-1-REIN-WCU-V1 can be interfaced with external host controller unit for example AS260-VCU-V1 or AIC-V2. The host can communicate with AS049-1-REIN-WCU-V1 via connector J1 and can power up AS049-1-REIN-WCU-V1 via power connector J11.
Below figure shows the stack up connection of AS049-1-REIN-WCU-V1 and AS260:

Figure 4-10 AS049-1-REIN-WCU-V1 and AS260 stack up.

5. Testing AS049-1-REIN-WCU-V1 board with AS260 Sample Software

The sample software of AS260 has Battery Management solution algorithms and cloud connectivity over Wifi (DA16200), BLE (DA14531) and LTE(EC25) implemented on different cores of RH850/U2A8 microcontroller in AS260_VCU_V1.

AS049-1-REIN-WCU-V1 is stacked over AS260_VCU_V1. And AS260 is also interfaced to AS261.

The sample software performs calculation of BMS data for 16 cells and upload the relevant readings and faults to the cloud for further cloud computation. The BMS algorithm is done by core0, and cloud connectivity is done by core1 of RH850/U2A8 MCU. The data is shared between the cores using shared memory called Cluster RAM.

The Wifi module DA16200 on AS049-1-REIN-WCU-V1 is configured as HTTP client and the data is uploaded to HTTP python-based server. The BLE module DA14531, is BLE peripheral device and configured as serial profile to upload the data to the smartphone. The LTE module EC25, is connected to internet via SIM network and the data is uploaded over cloud over MQTT protocol. EC25 acts as MQTT client and is connected to “broker.mqttdashboard.com” with port 1883.

5.1 Prerequisites for testing the Sample Software.

Sample Software folder

Extract AS260VCUSamplesoftware.zip in the desired folder which has sample software project AS260_AS261_REIN_WCU_DualCore.zip and http server application folder ‘web’. The details of these are given in further sections.

Hardware

1. AS049-1-REIN-WCU-V1 wireless connectivity board.
2. AS260-VCU-V1 board with harness
3. E2 emulator
4. AS261 Scalable BMS Companion Board (CB) and AFE boards.
5. Battery pack

Software Tools

1. [CS+](), [Renesas Flash programmer]().
2. Python installed on PC: [Python Releases for Windows | Python.org]() .

   Note that while installing the Python, you should check “Add python 3.10 to PATH” to add the environment variable to your PC, and select “Customize installation”, check the “pip” to install the pip together for the “flask” installation in the next step.

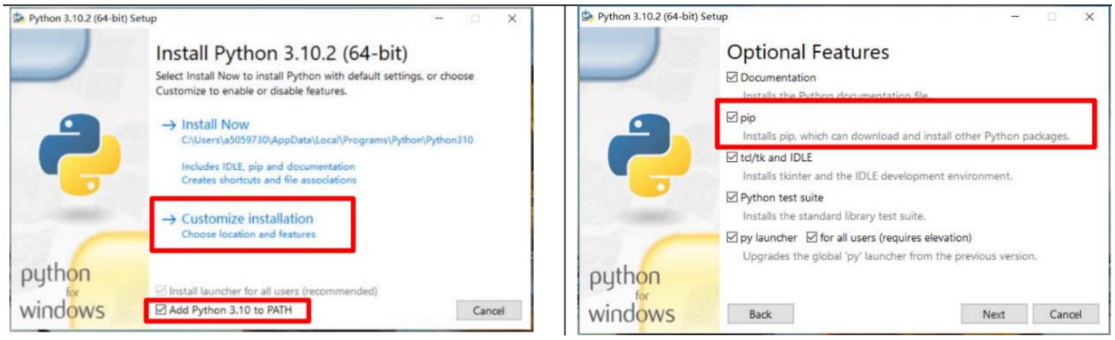

Figure 5-1 Installing Python

Open the “Command prompt” window, input “pip install flask” command to install flask. After installing it successfully, the following information will be shown.

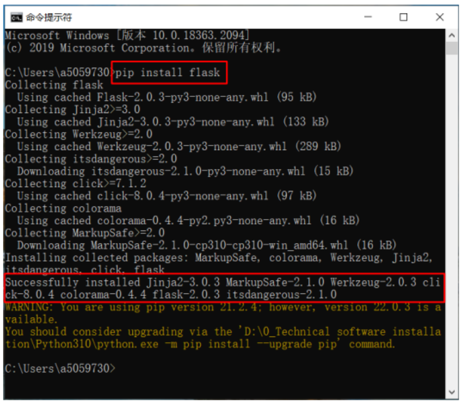

Figure 5-2 Installing flask

3. Smart Console mobile application installed on smart phone. Refer [SmartConsole Android and iOS Application — DA145XX Tutorial SDK Getting started (renesas.com)]() for details on application download and usage.
4. Hivemq websocket MQTT client, [MQTT Websocket Client (hivemq.com)](). Or any MQTT based client application connected to “broker.mqttdashboard.com” with port 1883.

5.2 Testing the Sample Software

To open the project, double click on u2a8_startup.mtpj in the below project path : U2A_16_Channel_code\U2A Startup Routine APN Rev1.1+MOSC=20MHz\Program Name\project\U2A8\u2a8_startup_cs+.

Once the project is opened you should set the server's name for HTTP connection in the project. The software package comes with for 'web' which has python application for HTTP server.

Locate the path to the "web" folder for HTTP server setup, for example, D:\web in Command Prompt. Input "httpserver.py" to run it. Then the HTTP server sets up. You can find the server IP address after the command run. For example, the HTTP server IP is http://192.168.8.21:5000 in the below figure. Press "Ctrl + C" to stop the HTTP server if need.

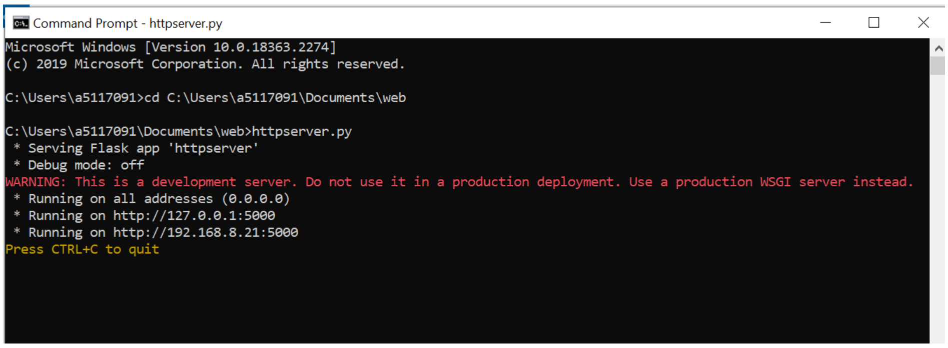

Figure 5-3 Run the HTTP server on PC.

You should set the IP address to the MCU source code, rebuild it, and then download it again to match your environment. In the project opened in CS+, navigate to hjp_wc.c in wc folder of PE1 core project. The edit the string "server_name[]" as per the server IP, as shown in the below figure and rebuild the application.

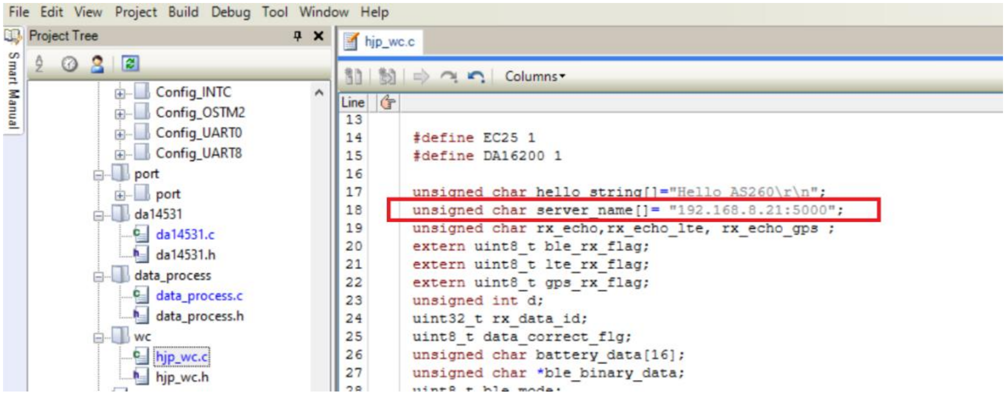

Figure 5-4 Edit the server IP

To program the board, follow the steps mentioned in section **Error! Reference source not found.**., program the 'u2a8_startup.mot' from DefaultBuild_merged folder.

For debugging follow the steps mentioned in the section **Error! Reference source not found.**.

Once programed, turn off the power, remove the E2 debugger and make the connections with AS261 and AS049-1-REIN-WCU-V1 WCU. Follow below steps while connecting the boards:

1. Connect AS260 to CB board of AS261 using I/O connector as shown in **Error! Reference source not found.**.
2. Stack up AS049-1-REIN-WCU-V1 to AS260 using Wireless board connector as shown in **Error! Reference source not found.**
   Make sure that SIM card is inserted in SIM slot and, Main and Div antennas are connected on the AS049-1-REIN-WCU-V1 as shown in below figure. Also configure the SIM as per the network provider and country region.

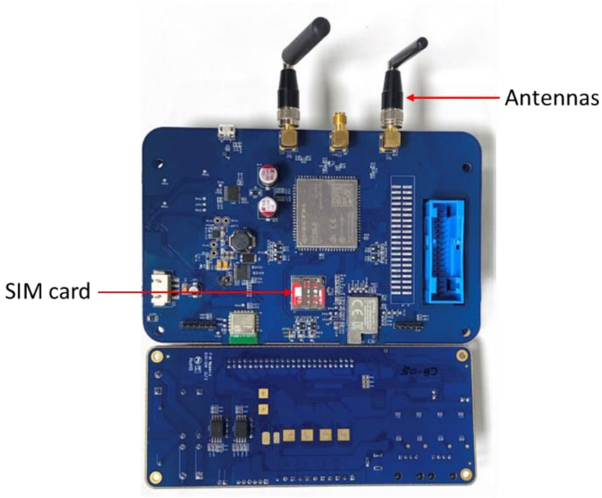

Figure 5-5. Antennas and SIM card on AS049-1-REIN-WCU-V1

3. Connect the AFE board and battery stack to CB.
4. Connect the 12V adaptor to each AS260 and CB board.

Below figure depict the flow of the connections followed in entire system.

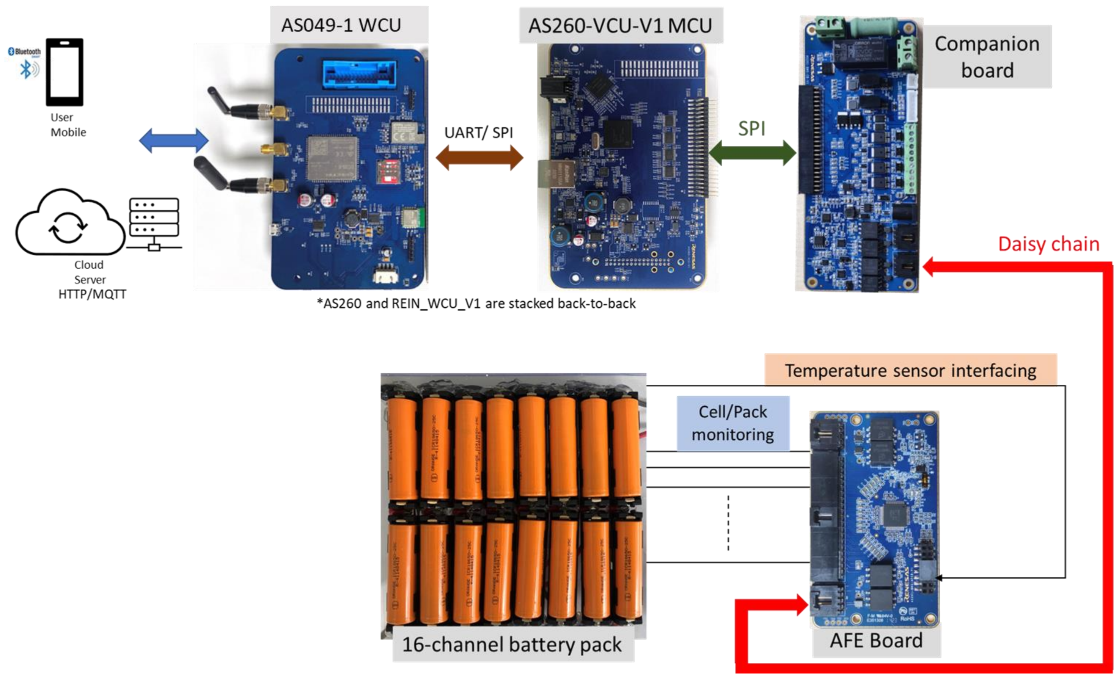

Figure 5-6 System Connection flow

Open the below applications on the respective devices for connecting respective modules:

1. Smart Console in smart phone.
2. Connect PC and DA16200 to same WiFi. Or turn on the hotspot function on PC, and then connect DA16200 to same hotspot.
3. Open HiveMQ WebSocket MQTT client on the browser.

Turn ON the boards. All the wireless modules will be initialized and be ready for the connection.

5.2.1. BLE (DA14531 module) Connection and Data Transfer

1. The Smart console will show 'Clv2-Codeless' in the list as shown below.

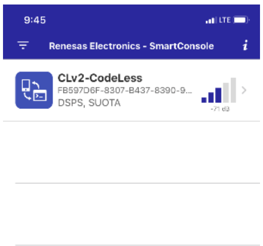

Figure 5-7. Smart Console detecting BLE device

2. Click it to connect to BLE device. The App will show Command mode window on it.

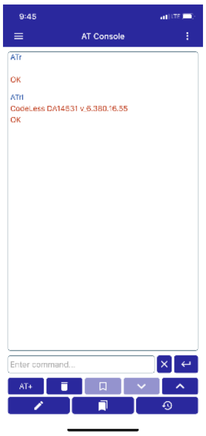

Figure 5-8. Command mode Window

3. After few seconds it will prompt Binary Request Received as shown below. Click on accept.

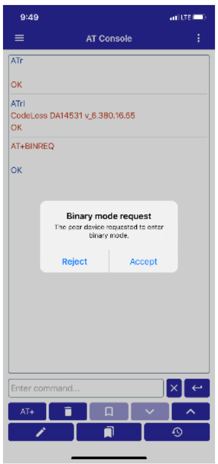

Figure 5-9. Binary mode request

4. After few seconds the data will appear as below on Smart console.

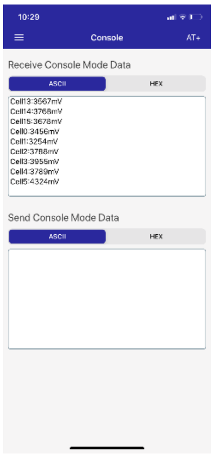

Figure 5-10. Data Reception on BLE

5.2.2. Data Upload over HTTP using Wifi Module

AS260 communicates with DA16200 on AS049-1-REIN-WCU-V1 which acts as HTTP client and uploads the data to the PC based server. HTTP server will show the data as below.

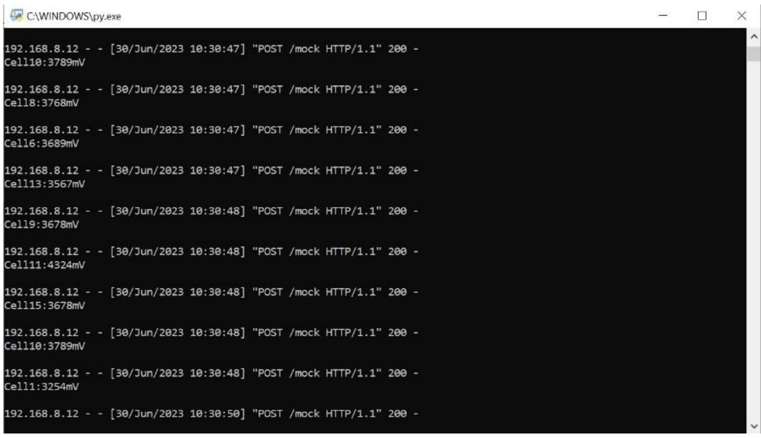

Figure 5-11. Data received on HTTP server

The data is display as 'Celln : xxxxmV', where n= cell number and xxxx is the cell voltage value in mV. For eg. If cell number 1 (n = 1) has 3254mV voltage (xxxx= 3254), the data displayed is as Cell1:3254mV.

5.2.3. Data Upload over MQTT using LTE Module

On HiveMQ client (or any MQTT client), connect to the broker by clicking on 'Connect' below:

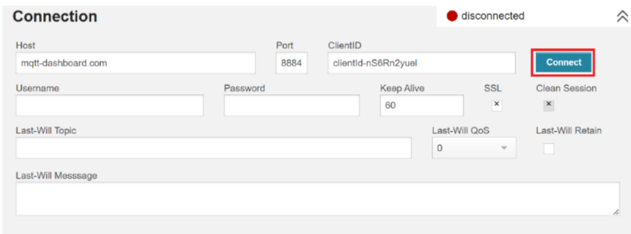

Figure 5-12 Connect HiveMQ client to the broker

Click on Add new subscription topic as shown below:

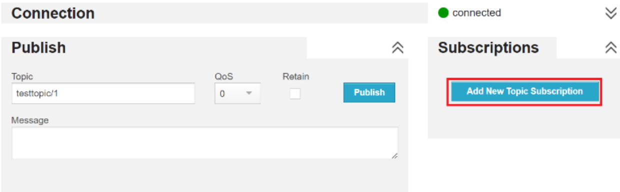

Figure 5-13. Add new subscription

Subscribe the topics, BMS/cell1, BMS/cell2…. BMS/cell16 as below.

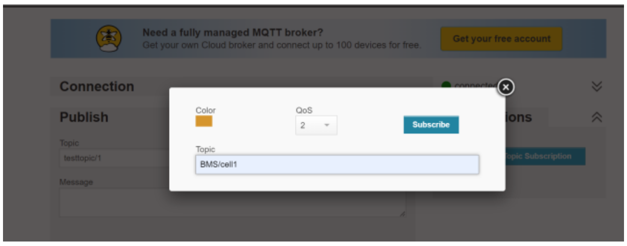

Figure 5-14 Subscribing topic in Hivemq WebSocket client.

Thus all 16-cell data will be uploaded and shown on the HiveMQ client. Below image shows six cell data uploaded in mV.

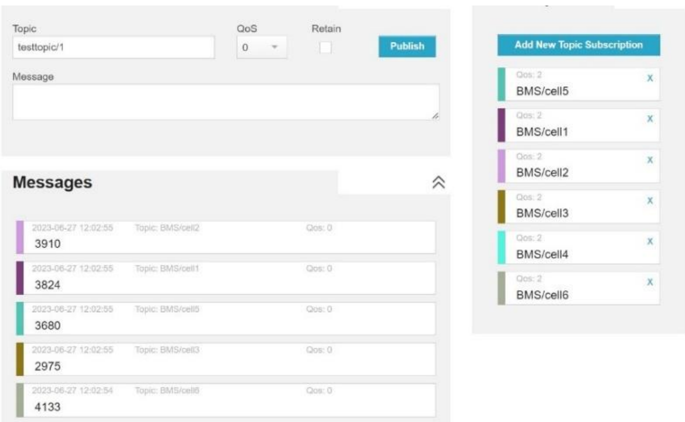

Figure 5-15. Cell Data uploaded over MQTT

Below is an example of IoT ON-OFF application on ios platform, which is MQTT client connected to “broker.mqttdashboard.com” with port 1883. This application shows graphical representation of the cell voltages received.

Figure 5-16. Graphical representation of Cell voltages vs time on IoT ON-OFF app

6. Acronyms and Abbreviations

Acronyms and Abbreviations	Explanation
BMS	Battery Management System
BLE	Bluetooth Low energy
HTTP	Hypertext Transfer Protocol
LTE	Long-Term Evolution
VCU	Vehicle control Unit
WCU	Wireless connectivity Unit
Wifi	Wireless Fidelity
MQTT	MQ Telemetry Transport

7. Revision History

Revision	Date	Description
1.00	June 30 2023	Initial release.